Learning Rich Hidden Markov Models in Document Analysis: Table Location

Ana Costa e Silva.
University of Edinburgh
Ana.Costa.e.Silva@ed.ac.uk

Abstract

Hidden Markov Models (HMM) are probabilistic graphical models for interdependent classification. In this paper we experiment with different ways of combining the components of an HMM for document analysis applications, in particular for finding tables in text. We show: a) how to integrate different document structure finders into the HMM; b) that transition probabilities should vary along the chain to embed general knowledge axioms of our field, c) some emission energies can be selectively ignored, and d) emission and transition probabilities can be weighed differently. We conclude these changes increase the expressiveness and usability of HMMs in our field.

1. Introduction

As explained in [7], to locate tables in documents only a few authors have used data induced learning algorithms: probabilistic modelling ([8]), Naïve-Bayes classifier ([4]), Maximum Entropy classifier ([4] and [5]), decision trees ([4], [6], [9]), SVM classifiers ([9]), the Winnow classifier ([4]), Conditional Random Fields [5] and an HMM of the traditional sort [5]. The approach followed by [6] was to use a decision tree to classify each non-empty line as belonging to a table or not. In this task over 90% precision and recall were achieved. Using a heuristic to join potential table lines into tables resulted in a high degree of completeness[1] (>90%) but low purity[2] (5%). We aim at increasing purity with little cost in completeness. We believe this can be done if the classification of each node coordinates with that of its neighbours. HMMs were devised to do just this. The question we try to answer is how to best combine the components of an HMM to capture the specificities of document analysis.

[1] *Completeness* is the ratio between the number of detected tables that contain all the lines that belong to real tables by the number of tables that exist in the dataset.

[2] *Purity* is the ratio between the number of detected tables that contain only the lines of the corresponding real table by the number of tables detected by the algorithm.

2. Overview of HMMs

Figure 1 shows the typical representation of an HMM. For each node the value x_i is observed; we wish to determine z_i. Each z_i depends on: the classifications of its neighbour z_{i-1} (this dependency is modelled in the transition table $P(z_i|z_{i-1})$); and on what we observed about it (modelled in the emission table, $P(x_i|z_i)$).

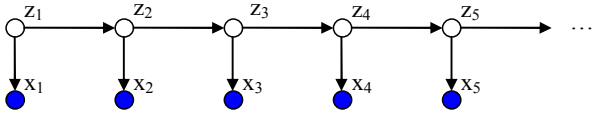

Figure 1. Typical representation of an HMM

By inspecting the graph, we can write the joint probability function for the entire chain of nodes as:

$$P(x_1...x_n, z_1...z_n) = P(z_1) * \underbrace{\prod_{i=2}^{n} P(z_i|z_{i-1})}_{\text{Transition effect}} * \underbrace{\prod_{i=1}^{n} P(x_i|z_i)}_{\text{Node effect}} \quad (1)$$

The transition effect is generally simple to model by simply computing how often subsequent nodes have different classifications. The node effect is often less simple to capture. [1] suggests that by direct application of Bayes theorem to equation 1 and because x_i is observed, we obtain

$$P(x_1...x_n, z_1...z_n) \propto P(z_1) * \prod_{i=2}^{n} P(z_i|z_{i-1}) * \prod_{i=1}^{n} \frac{P(z_i|x_i)}{P(z_i)} \quad (2)$$

where $P(z_i)$ is the prior probability of each class in the whole dataset, and $P(z_i|x_i)$ is the proportion of each class in any grouping of x_i. This is very interesting because we can approximate the node effect, for any identifiable subset of data, by simply taking in it the proportion of each class, e.g. if 1000 lines look like titles and 2 are in fact table lines, then $P(z_i \mid x_i)$= 0.2%.

3. Design of the experiment

The goal of our experiments is to group document lines into tables (we mean non-empty lines, separated by any number of empty lines). For this, we will try different ways of combining the transition and node effects. In each approach, only one change is made with respect to the previous and results are compared. Each change aims at answering one question.

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.185

Model 0 is our baseline, [6]. It uses a decision tree for finding table lines and a heuristic for grouping them into tables. It does not use HMMs.

But does using an HMM improve our ability to locate tables? Model 1 uses a traditional HMM, where the way each node affects its neighbours is the same along the whole chain, and the way each observed value affects each node is derived from the decision tree in [6], which gives $P(Z_i|X_i)$. X_i simply takes value "looks like table line" / "does not look like table line".

But does adding knowledge on more document structures improve results? Model 2 includes classifiers for titles, headers and footers, charts, text in one or two columns; as such X_i also can take values as "looks like text", "looks like chart", looks like title". We combine these classifiers to estimate the HMM's node effect, as explained in 4.2. We hope the HMM will elegantly combine them into one consistent page configuration.

But does it make sense to have transition tables that are the same along the whole chain / page? Model 3 changes the way the transition effect is captured. In fact, we noticed that, for example, when more empty lines separate two lines, the probability of there being a change in document structures increases. This information must be captured by the model's transition probability – not doing so would average over rather disparate realities. This idea is however rather novel.

But does it make sense to systematically feed the HMM with all we know about a node? Specifically, the decision tree had classified 10,521 lines as non-table (leaves 4 to 6 in Table 1) and got 803 errors wrong. 803 is a large absolute number of errors, which enter the HMM with very high emission strength. For these nodes, in Model 4 we simply make $P(Z_i|X_i)$ equal the prior probability $P(Z_i)$, which, as captured by equation 3, switches to 0 their node effect.

But could breaking the constraint that transition and node effects are equally important improve results? In Model 5, we changed the relative strength with which the node and the transition effects enter into the joint probability. This corresponds to using equation 3:

$$P(x_1 \ldots x_n, z_1 \ldots z_n) \propto \left[P(z_1) * \prod_{i=2}^{n} P(z_i|z_{i-1}) \right]^{\alpha} * \left[\prod_{j=1}^{n} \frac{P(z_i|x_i)}{P(z_i)} \right]^{\beta} \quad (3)$$

With this formulation, one can model non-linear implications of the node and transition effects over the joint probability. α and β should be in trade-off, e.g. α+β=1. We tuned α by approximation, by taking several tentative values between 0 and 1 and chosing the one that provided better results. We have done this based on models 3 and 4, obtaining in model 5.1 and 5.2 respectively. For Model 5.1, the best performing values of α and β were 0.34 and 0.66 respectively (not very different from one another), while for Model 5.2 they were 0.32 and 0.68 respectively, which suggests that, for our data, the transition effect should have much smaller importance than the emission effect.

Finally, we will determine how to best combine the components of an HMM for table recognition, by comparing the performance of these models.

4. Preparatory work

4.1. Description of our dataset

To ground our experiments, we gathered 22 pdf financial statements, which we converted to ASCII with the pdftotxt linux utility. This context is particularly demanding, as financial tables tend to be very varied, with unequally filled-in columns and rows and complex multi-line headers. Our documents have lengths varying between 13 and 235 pages that contain between 3 and 162 tables. We randomly chose 19 documents for training and 3 for testing.

In this work, we only use the subset of our dataset belonging to pages that contain at least 3 table-looking lines (as chosen by the decision tree in [6]). There-in we exclude lines that we identified as headers of footers, using a strategy inspired by [3]. As such, our training (test) set has 29,217 (3,043) non-empty lines, of which 60% (70%) belong to tables.

4.2. Estimating the node effect

The emission probabilities used in HMMs depict how likely an observed value X is for each class of Z, $P(x_i|z_i)$. This can be hard to model. As explained in section 2, we estimate the node effect with $P(z_i|x_i)$.

In [6], we had used a decision tree to classify each line as belonging to a table or not. It distinguished six leaves; each leave is a value of x_i. $P(z_i|x_i)$ is calculated as the proportion of each class in each of the six leaves. Only these, which we present in the top part of Table 1, will be used to estimate the node effect in Model 1.

However, "often […] improved performance can be obtained by combining multiple models together in some way, instead of just using a single model in isolation", [2]. We decided to search for subsets of the document – headers or footers, charts, titles or text – where we are almost certain of their classification, so as to better ground overall results. Our dataset is not manually annotated, so we used rules to identify these.

Identifying titles: We created a parser for finding numbered or regular titles. It finds 1,229 titles with 98% of being non-table, i.e. $P(z_i=0|\ x_i=\text{"title"})$=98%.

Finding charts: 478 lines with $P(z_i=0|x_i)$=78.4%.

Finding text organised in one column: Within each document, we characterise its typical one-column line by, from lines with no interior spaces, taking the mode

of the number of non-space characters. We compare all lines to it: DifLen = Number of characters in a line minus that Mode. In Table 1, we see that, when DifLen is closer to 0, we are more unlikely to have a table line.

Finding text organised in two columns: one can infer whether a document is generally organised in one or more columns by taking the mode of interior spaces per line (when more than 0) – one column documents tend to have a mode of 2 or 4, while multicolumn documents will present a bigger mode.

In multicolumn documents, we select all lines with as many interior spaces as the document's mode +/- 6. From these lines, we compute the mode of the number of letters per line and the mode position of the first white space gutter. We then compute MaxDif as the maximum difference of each line to these modes. Table 1 shows how $P(z_i=0|x_i)$ changes with MaxDif.

Table 1. Conditional probability table, $P(z_i=0|x_i)$

	X_i="Line looks only like…"	X_i="Line looks like… but also looks like a table"	
		Leaf 1	Leaf 2
Looks like...	$P(z_i=0\mid x_i)$		
A table, Leaf 1	11.9%	--	--
Leaf 2	19.2%	--	--
Leaf 3	42.3%	--	--
Leaf 4	81.0%	--	--
Leaf 5	87.1%	--	--
Leaf 6	94.7%	--	--
A chart	78.4%	76%	84.0%
A title	98.0%	98.7%	
A text line in a page organised in one column,			
DifLen >=-5	94.2%	91.8%	--
DifLen <- 5	83.5%	38.8%	--
DifLen <-15	59.4%	--	--
DifLen <-25	50.5%	--	--
A text line in a page organised in two columns,			
MaxDif < 5	100%	100%	100%
MaxDif < 15	99.5%	100%	100%
MaxDif < 25	95.3%	89.1%	64.7%
MaxDif < 35	81.3%	76.1%	58.5%
MaxDif>=35	53.4%	--	--

Combining the different classifiers: Because we are interested in isolating tables, the probability $P(Z_i=0|X_i)$ when all experts agree on a non-table classification is simply the maximum value produced by all. But when for a line there is disagreement, in particular when two experts have a degree of certainty of disjoint classifications that is above the 60% prior probability, a combining strategy is required. We considered *tree-based models* (TB) and *mixture of experts* (ME), [2].

TBs divide the input space into regions that are homogeneous in terms of the variable we are trying to predict. Then a simple model, e.g. a constant, is assigned to each region. Hence, we should isolate lines with conflicting classification and computes $P(z_i=0|x_i)$ as the proportion of observed non-table lines in the set.

On the other hand, MEs calculate $P(Z_i|X_i)$ as the weighted average of the probabilities assigned by each expert, the weights (called gating coefficients), often being learnt from data. However, since they express how much we trust a given expert to make predictions in an area of the input space, we can measure this trust by the proportion of times each expert gets a prediction correct, which is precisely the probability used in TB.

When comparing the probabilities calculated under both models, they are either rather close or ME gives much higher values of $P(Z_i|X_i)$. We think that, if different experts claim clashing classifications for an observation, this is as an added factor of uncertainty and so favouring an approach that gives smaller probabilities is cautious: we thus prefer TBs.

In Table 1, we can see the final conditional probability table, which, divided by the prior of 60%, will approximate the emission probability in equations 2 and 3. We will apply it in Models 2 and 3. In Model 4 and 5, as explained in section 3, the first two lines will be replaced by 40%, which is $P(Z_i=0)$.

4.3. Estimating the transition probability (TP)

Generally the TP used in HMM is stable, meaning that it is the same throughout all the chain. This is most suitable for several of the traditional applications of HMMs, such as recognition of sound (music or text), linguistic sequences (e.g. part of speech tagging), molecular biology, or for tracking objects in video. In these fields, nodes tend to occur one after the other in a stable way. To estimate a T.P. table under this approach, one simply counts the proportion of times a non-table line is followed by a table line and v.v. In our sample, non-table lines (z=0) are followed by table lines (z=1) 7% of the times, and 5.3% of the vice-versa. Model 1 and 2 use these numbers as their TP. However, we believe this not to be the best way of estimating a TP for document analysis tasks.

The effect of distance between lines. It is intuitive that if two lines are separated by more white space, they are more likely to belong to distinct document structures. We measure DisLin as the number of empty lines between two non-empty lines, plus one. In our data, two consecutive lines (DisLin=1) have only 3% probability of being of different structures; but when DisLin>5, a table line is followed by non-table in 86% of cases. A TP that does not account for this effect averages over rather disparate realities!

The effect of difference in left alignment. Another intuition of document analysts is that a change in document structure is often accompanied by some form of alignment modification. In fact, when between two adjacent lines the difference in the distance to the left margin of the page (DifLef) is bigger than 20 (150) spaces, there is a 27% chance of them flipping types. When DifLen=0, however, that chance is only 1%.

The joint effect of distance between lines and difference in left alignment. In Table 2, we present the transition probability table that we will use from Model 3 onwards. The last two rows read: when the distance between two lines is above 4, or when it is bigger than 3 and the difference in left alignment is more than 150 spaces, the probability of a table ending is 84.7%.

Table 2. Transition probability table

Probability of non-table following a table line when…		
DifLef <= 20	And DifLin = 1	1.0%
DifLef = 0	And DifLin > 1	5.7%
DifLef > 20	And DifLin = 1	13.3%
DifLef <= 20	And DifLin > 1	
DifLef > 20	And DifLin > 1	48.4%
Probability of table following a non-table line when…		
DifLef = 0	And DifLin = 1	0.4%
1 < DifLef <= 150	And DifLin = 1	2.5%
DifLef = 0	And DifLin = 2	
DifLef > 150	And DifLin = 1	10.3%
1 < DifLef <= 100	And DifLin = 2	
DifLef = 0	And DifLin = 3	25.0%
100 < DifLef <= 150	And DifLin = 2	
DifLef = 0	And DifLin = 4	38.0%
1 < DifLef <= 100	And DifLin = 3	
DifLef > 150	And DifLin = 2	
1< DifLef <= 100	And DifLin = 4	71.7%
100 < DifLef <= 150	And 2< DifLin < 5	
DifLef > 150	And DifLen >= 3	84.7%
	DifLen >= 5	

5. Our results

In this paper we evaluate alternative ways of using an HMM in a document analysis problem, in particular for locating tables in ASCII documents. Each successive model is more complex than its predecessor. Our analysis here is intended at demonstrating the effectiveness of each added component. In Figure 2, we present completeness, purity and their harmonic mean, which we call CPF. With it, we can answer the questions of section 3.

When compared with Model 0, Model 1 makes clear that using an HMM for table location is a good idea. Model 2 shows that supplying the HMM with information from different document structure detectors also improves results. Model 3 suggests that using intelligent adaptive transition tables that incorporate generally available document knowledge is an added plus. In Model 4 we had restated downwards the emission estimates of nodes that contained numerous mistakes. For a user who likes completeness and purity equally, it may not represent a good choice, since CPF improves in the training set but not in the test set. However for one who prefers completeness, model 4 is better. Model 5.1 shows that adding different transition and emission weights to Model 3 reduces CPF. Finally, when in Model 5.2 different weights were added to Model 4, CPF improved in both datasets. Model 5.2 finds the biggest number of complete and pure tables; its completeness is similar to Model's 0 but its purity is over 4 times higher.

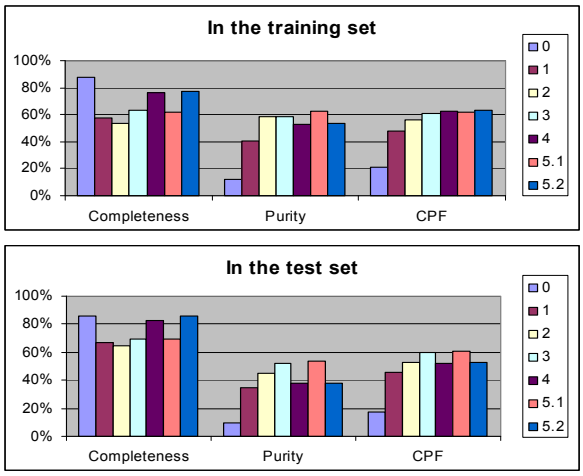

Figure 2. Performance evaluation

6. Insufficiencies of the HMM

After our modelling work, we noticed that, although our favoured HMM can, much better than ever before, group potential table lines into tables, there are still aspects that can not be adequately modelled. For example, we know that the first and last lines of a table rarely contain only one cell of left aligned text – 99% of tables in our dataset follow this rule. We obviously could not express this in the emission probability because left aligned one cell lines are common within tables and text, they are just uncommon in transitions. So we adapted the transition table by inscribing really low probabilities (which we learnt from data) in these lines. This has worsened results. We believe this happened because sometimes a line that cannot be a transition encompasses transition signs that warn of a transition in the area, for example in the next line. We found, therefore, that it is more appropriate to verify

and impose these and other constraints on the end result of our model of choice.

On the other hand, we noticed that, given the way we formulated our HMM, two different tables get glued together if between them there is no non-table like structure. We decided to treat also these cases *a posteriori*. In particular, in Table 2 it is apparent that when two lines are separated from each other by more than 5 lines OR more than three and have a difference in left alignment of more than 150 spaces, there is an 84.7% chance they do not belong to the same table; these odds are 65.1% for lines that distance more than 4 lines and have at least some difference in left alignment. Any time we notice such patterns inside a table, we assume we found a divisor between two different document structures. If above and below this divisor there are at least 3 lines, we assume there are two tables (no table can ever have less than 3 lines).

We implemented these constraints to model 4 and 5.2 and measured results. Purity increases to above 60% with little cost in completeness in both models and on both datasets; CPFs increases accordingly. As discussed, our dataset is particularly difficult – a method doing well here will do better in most tables.

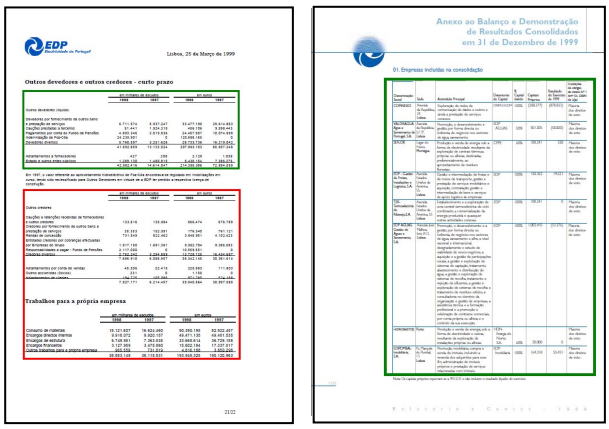

Figure 3. Examples of failure and success

In Figure 3, we show two real pages. On the left, the method succeeded in handling horizontally spanning cells and at cutting out titles and text, but it failed to unmerge the two bottom tables since, in ASCII, the heading of the bottom-most looks like a cell's content. On the right, the method works fine with vertically spanning cells and rightfully remove's the table's footnote and title.

7. Conclusion

In this paper we have shown that:

- HMMs can be used with benefit to locate tables;
- the probabilities that go into the HMM can be straightforwardly learnt from data;
- different document structure detectors, derived independently, can be optimally combined, HMM elegantly balancing them on the page;
- adaptable transition tables considerably increases our ability to express document analysis patterns;
- weighing the transition and emission probability differently allows further flexibility in modelling complex probability distributions in a simple way.

Finally, we have seen what aspects of our modelling needs are not expressible with HMM. Interesting future work would be to find a way of adding such constraints into an equally elegant probabilistic formulation.

8. References

1. Bishop, Christopher M., Markus Svensen and Geoffrey E. Hinton, "Distinguishing text from graphics in on-line handwritten ink", IWFHR, Japan, 2004.
2. Bishop Christopher M., "Pattern recognition and machine learning", Springer, 2006.
3. Chao, Hui, "Background pattern recognition in multi-page PDF document", DLIA, Edinburgh, 2003.
4. Cohen, W.W., M. Hurst, L.S. Jensen, "A flexible learning system for wrapping tables and lists in HTML documents, 11 WWW Conference, USA 2002.
5. Pinto, D., A. McCallum, X. Wei, W.B. Croft, "Table extraction using conditional random fields", SIGIR, Canada, 2003.
6. Silva, Ana Costa e, "New metrics for evaluating performance in document analysis tasks - application to the table case", ICDAR, Brazil, 2007.
7. Silva, Ana Costa e, Alípio Jorge, Luis Torgo, "Design of an end-to-end method to extract information from tables", IJDAR 8(2), Special issue on detection and understanding of tables and forms for document processing applications, 2004-06, pp. 144-171.
8. Wang, Y. (2000-02), "Document analysis: a table structure understanding and zone content classification", PHD Thesis, Washington Univ., USA.
9. Wang & Hu (2002), "A machine learning based approach for table detection on the web", International WWW Conference 11, USA 2002, pp. 242-250.